

loading a content into the identified region; and
[registering an identity of the content of the secured region; and]
causing the one processor to jump to a known entry point in the content.

2. (Once amended) The method of claim 1, further comprising:
preventing interference with the identifying, loading, and registering by [each] at least one
of a remaining one of the plurality of processors.
3. (Once amended) The method of claim 2, wherein preventing interference comprises
halting [each] at least one of the remaining ones of the plurality of processors until the
identifying, loading, and registering is complete.
4. (Once amended) The method of claim 2, further comprising:
causing [each] at least one of the remaining ones of the plurality of processors to jump to
the known entry point in the content.
5. (Once amended) The method of claim 1, wherein identifying comprises receiving a region
parameter, the region parameter specifying a location of the region.
6. (Once amended) The method of claim 5, wherein the location [is] comprises a range of
addresses in the memory of the computer within which the region is located.
8. (Once amended) The method of claim 1, further comprising:
registering an identity of the content of the identified region, wherein registering
comprises:

recording a hash digest of the content of the [secured] identified region; and
signing the hash digest, the signed hash digest being stored in a register in the memory of
the computer that is accessible to a third party to verify whether the content can be trusted.

32. (Once amended) An apparatus to load a trustable operating system comprising:

a first processor having a start secure operation (SSO), the SSO having a memory region
parameter, wherein the first processor [executes] is capable of executing the SSO to block access
to a region of memory specified in the memory region parameter and to place a content in the
specified region;

a hash digest, wherein the first processor further [executes] is capable of executing the
SSO to erase a current content of the hash digest and to record in the hash digest a cryptographic
hash of the content of the specified region; and

wherein the first processor further [executes] is capable of executing the SSO to unblock
access to the specified region and to jump to a known entry point in the content of the specified
region.

33. (Once amended) The apparatus of claim 32, further comprising:

a second processor, the second processor having a join secure operation (JSO), wherein
the second processor [executes] is capable of executing the JSO to prevent the second processor
from interfering with the first processor's execution of the SSO.

34. (Once amended) The apparatus of claim 33, wherein the second processor is capable of
commencing[es] execution of the JSO when the first processor commences execution of the
SSO.

35. (Once amended) The apparatus of claim 33, wherein, to prevent the second processor from interfering with the first processor's execution of the SSO, the JSO is capable of causing[es] the second processor to enter a halted state until the first processor's execution of the SSO is complete.

36. The apparatus of claim 35, wherein the first processor [executes] is capable of executing the JSO to further cause the second processor to exit the halted state after the first processor's execution of the SSO is complete and to begin executing at the known entry point in the content of the specified region.

37. The apparatus of claim 32, further comprising a digest signing engine having a secure channel to access the hash digest, the digest signing engine capable of computing the cryptographic hash of the content in the specified region in response to a request by the first processor executing the SSO.

Please add the following new claims:

39. A method of loading a trustable operating system comprising:
selecting an area in a memory accessible to a processor;
loading a data into the selected area;
directing the processor to commence processing at an entry point in the selected area; and
preventing interruption of the selecting, loading, and directing until they are completed.

40. The method of claim 39, wherein preventing interruption comprises halting any other processors having access to the memory until the selecting, loading, and directing is complete.

41. The method of claim 40, further comprising:
causing the other processors to commence processing at an entry point in the selected area.
42. The method of claim 39, wherein selecting comprises receiving a parameter specifying a location of the area to be selected.
43. The method of claim 42, wherein the location is a range of addresses in memory within which the area is located.
44. The method of claim 42, wherein the location comprises a start address and a length of memory within which the area is located.
45. The method of claim 39, further comprising
registering an identity of the data loaded in the selected area;
recording a unique cryptographic function of the data loaded in the selected area; and
signing the unique cryptographic function, the signed unique cryptographic function
being stored in a register in memory.
46. The method of claim 39 wherein the data is a component of an operating system to operate a device in which the memory resides.
47. The method of claim 46, wherein the operating system has a graphical user interface.
-